Syllabus

Production Quality Computational Multiphysics Software Development

A University of Colorado Boulder Distance Learning Course

Course No. CSCI 7000-200B (for credit), NCEN 5838-570 (for non-credit)

Spring, 2023 (January 18 - May 3, 2023)

Monday-Wednesday-Friday 12:00pm-12:50pm, ECCS 1B12

Instructor: Scott R. Runnels, Ph.D. ECOT 411
scott.runnels@colorado.edu 505-695-9241

Contents

1 Course Description

Multiphysics code development is a multidisciplinary endeavor, which by its nature requires contributors who have different strengths. At the same time, each contributor is eventually required to perform beyond their trained area of expertise. In this course, you will be exposed to some of the most essential elements of multiphysics code development, in such a way that builds your knowledge, skills, and confidence both for your current research efforts and your possible future role in the field of computational physics. The key multidisciplinary components include numerical methods for discretizing time-dependent partial differential equations, numerical analysis, linear and nonlinear solvers, computer programming, software verification, version control, production process control, testing, and the underlying physics being simulated. This course provides an in-depth survey of examples across all of these areas and provides a level of assurance to the successful student that they have enough knowledge to confidently participate in a multiphysics software endeavor in the future while also accelerating their own research now.

The course uses a combination of technical homework problems, software development assignments, and a project to instill understanding and develop skills. There is considerable flexibility in project choices. You may work alone or on a small team. You may start a project from scratch or use what you learn in this course to advance an existing code, e.g., your research. For every project choice, you will be required to implement multiple components, depending on their complexity and state of the code with which you start, that were not pre-existing in your work. Examples include: (1) A new discretization method, (2) A new solver method, (3) Self-documenting regression tests, (4) A test harness, (5) Verification testing, or (6) Repository management. You will be required to write a project pre-proposal, proposal, update, and final report. The goal here is to learn by doing, and to make the “doing” part as relevant as possible to your current research or production work.

2 Learning Outcomes

1. Experiencing live code development for these common methods:
   - Finite element method for elliptic and parabolic systems
   - Finite volume method hyperbolic systems, shock physics
   - Finite difference method for diffusion and electromagnetics
   - Time marching schemes
   - Nonlinear solvers
   - Iterative linear solvers
2. Hands-on experience in augmenting codes for:
   - Verification testing using the method of manufactured solutions
   - Multiphysics nonlinear coupling
   - Boundary fitting, curvilinear discretizations
3. Understanding of common analytics for numerical methods, including:
   - Theory of spatial and temporal convergence
   - Stability of diffusive systems
   - Stability of hyperbolic systems
4. Experience in basic tools of production code development, including:

- Verification using the method of manufactured solutions
- Version control using Git
- Continuous Integration (CI)
- C++ object-oriented programming
- Python
- Developing a regression testing system
- Developing automatic documentation for testing
- Introducing unit testing into a multiphysics code

5. Career enhancement through these motivational lectures:

- Using communication to stabilize and advance one's career
- Conducting effective peer reviews
- The philosophy of test-driven code development
- Time estimation

3 Prerequisites

This is intended to be a first-year graduate level course, and does not have any formal prerequisites. The field of computational multiphysics is multidisciplinary, and people who enter it have different strengths and weaknesses depending on their backgrounds. Those types of differences among the participants in this course are expected. If you have questions about your ability to participate in this course productively, please feel free to contact the instructor to discuss it. The following is provided as a guideline.

The theory lectures are designed such that they do not assume much knowledge beyond undergraduate mathematics for technical degrees. However, the lectures do quickly build upon that knowledge and so you should be firmly grounded in your undergraduate mathematics, including Calculus (I-III), with vector calculus, and you should understand what differential equations are. At least some previous exposure to partial differential equations is also very valuable, and can possibly be considered required. If you have not had any exposure to partial differential equations, you may not be ready to take or audit this course. Contact the instructor to discuss it first. Exposure to linear algebra will be very helpful. You should also have some experience programming (in any language); without it, the assignments will take you a lot longer. It is still doable as a new programmer because C++ and Python tutorials are included, but only if you are willing to take on the task of learning how to program while taking this course.

This course will deal with physical laws, in particular, the idea of "conservation", e.g., conservation of mass, momentum, and energy. Physics and engineering courses that provide students with knowledge of these laws are not required as prerequisites. But if you have not taken those types of courses, you will benefit from doing some background reading on those topics first, especially relative to partial differential equations. Also, you would benefit from discussing your participation in this course with the instructor for a more specific evaluation and recommendation.

4 Grading Structure

Component	Portion of Grade
Homeworks	40%
Project Milestones	45%
Final Exam	15%

5 How to Enroll with a Non-Credit Option

Students, faculty, post-docs, and practicing professionals are all welcome to participate in this course under a non-credit option. The instructions for doing that are provided here. If you, instead, desire to register for credit, please follow the standard enrollment procedures at www.colorado.edu.

Technically, there is not an audit option *per se* for this course. Rather, it is a non-credit, pass/fail course where the requirement for passing is to attend or listen to 30% of the lectures. Participants will self-report at the end of the course regarding which lectures they saw or attended and will be on the honor system for that. To enroll with this non-credit pass/fail option, fill out the form at this link: Registration Step 1.

After you have done that, you will receive a “welcome email” that will provide further instructions, including instructions for how to pay for the course or have your employer pay for the course. If you run into any problems, feel free to contact the instructor at scott.runnels@colorado.edu.

6 Estimated Schedule and Course Content

Note that the schedule and content are subject to change.

6.1 Week #1 Foundations (Part 1) (week of 1/16)

In this first week, we establish common ground, e.g., a common language for the various in-class software development demonstrations.

During Class:

- Orientation, motivation, expectations
- C++ Tutorial: In-class object-oriented code development tutorial providing all syntax needed for the course
- Python Tutorial: In-class code development tutorial demonstrating tools needed for the course

Assignment (5%): HW 1 Programming basics

6.2 Week #2 Foundations (Part 2) (week of 1/23)

This week, we continue developing common ground for the course, including version control with git and a common language for the mathematics.

During Class:

- git Tutorial, Part 1: In-class demonstration (creation, add, commit, push, pull)
- git Tutorial, Part 2: In-class demonstration (branching, merging, local/remote, difftools)
- Math Primer, Part 1
- Math Primer, Part 2
- Review of Continuum Equations: (Reynolds Transport Theorem)
- Applications of Continuum Equations to Mechanics, Part 1: Fluids

Assignment (5%): HW 2 Math primer and governing equations

6.3 Week #3 Foundations (Part 3) (week of 1/30)

This week we continue building foundations for the course and, ultimately establish enough theory to write down our first computational physics algorithm, the finite difference method.

During Class:

- Applications of Continuum Equations to Mechanics, Part 2: Solids
- Electromagnetics (Maxwell's Equations)
- Physical Constitutive Models, Phenomenological Models
- Finite Difference Method
- Gauss-Seidel Linear Solver

6.4 Week #4 Regression Testing of a Single Physics Code (week of 2/6)

This week we establish a production computational physics environment by developing a working computational (single) physics code while placing it under our own, custom-developed regression testing system. Also, this week, students will submit a pre-proposal for their course rpoject.

During Class:

- In-Class Demonstration: Development of a steady-state 2-D finite difference code
- Introduction to Regression Testing
- In-Class Demonstration: Development of a regression test system
- In-Class Demonstration: Adding regression testing to Continuous Integration
- Temporal Discretization Methods (Backward Euler, Forward Euler, Crank-Nicholson, Runge-Kutta)

Project (10%): Written project pre-proposal

6.5 Week #5 Self-Documenting Regression Testing (week of 2/13)

This week we greatly enhance our production environment by expanding the regression test system to automatically build documentation that explains the tests, the expected solutions, and an evaluation of the code. We will build skills in two essential areas of a production environment: Planning and Peer Review. Also, students will submit a full project proposal, revised based on comments from the instructor.

During Class:

- In-Class Demonstration: Development of an implicit transient 2-D finite difference code
- Using Latex and Python for Automatic Documentation Deployment
- In-Class Demonstration: Development of a self-documenting regression test system
- *Production Essentials Lecture:* Development Planning and Time Estimation
- *Production Essentials Lecture:* Effective Peer Review Methods

Project (5%): Revised project proposal

6.6 Week #6 Preparing for Multi-Physics Code Development (week of 2/20)

This week, we begin making preparations for expanding our production computational physics software development environment to include multi-physics by deriving the equation that will accompany our existing diffusion solver. Also, students will present their project plan for a peer review this week. Students will be graded on participation, style of offering criticism, and style of receiving criticism. This aspect of the course is aimed at developing and exercising the "soft skills" essential on a production team and during client interaction.

During Class:

- In-Class Demonstration Continued: Implicit transient 2-D finite difference code
- Students present project plans for peer review

Project (10%): Project plan presentations and peer review.

6.7 Week #7 Multi-Physics and Introduction to MMS (week of 2/27)

This week is dedicated to handling the nonlinearites associated with multi-physics, including beginning to lay the ground work for another key element in a production environment: verification testing of multi- and single-physics applications.

During Class:

- Voltage equation, resistive heating
- Thermal-electric coupling, non-linear systems
- Neumann boundary conditions
- Non-Linear Solvers: Successive Iteration, Non-Linear Lagging, Newton-Raphson, relaxation

- Inter-relationship of numerical tolerances for non-linear coupled systems (linear solver, non-linear solver, temporal solver)
- In-Class Demonstration: Expanding a Code from Single- to Multi-Physics
- Method of Manufactured Solutions, or “MMS” for single-physics applications

Assignment (5%): HW 3 Nonlinear solver for multiphysics code

6.8 Week #8 Code Verification (week of 3/6)

This week we demonstrate how to use the method of manufactured solutions in a production environment, involving self-documenting regression testing. We also begin our study of other linear solvers.

During Class:

- MMS for multiphysics applications
- In-Class Demonstration: Addition of MMS to an implicit 2-D finite difference code with automation regression documentation
- *Production Essentials Lecture:* Test Driven Code Development
- Introduction to Multi-Grid, Algebraic Multi-Grid solver methods

Assignment (5%): HW 4 Adding MMS to a multi-physics code

6.9 Week #9 Other Linear Solvers (week of 3/13)

This week we continue to expand our knowledge of other types of linear solvers important in production computational physics. Also, students will provide a project status update (1 page) describing their progress towards their project goals.

During Class:

- In-Class Demonstration: Development of a multi-grid implicit 2-D finite difference code
- Conjugate Gradient Linear Solver, Part 1: Theory
- Conjugate Gradient Linear Solver, Part 2: In-Class Development
- Production Essentials Lecture: Documenting and Using the Release Process

6.10 Week #10 Other Discretization Methods: Finite Elements (week of 3/20)

This week we expand our knowledge base through a carefully developed and efficient introduction to the finite element method and the data structures that support it. The finite element method is an essential technology in many computational physics codes.

During Class:

- Finite Element Method Theory

- Finite Element Method Data Structures
- In-Class Demonstration: Initial development of an implicit 2-D finite element code

  Assignment (5%): HW 5 Finite elements theory

6.11 Week #11 Spring Break (week of 3/27)

Enjoy your break!

During Class:

- No lectures

6.12 Week #12 Other Discretization Methods: Mimetic and Finite Volume (week of 4/3)

Multiphysics often involves multiple discretization methods. This week we complete the finite element code but also branch out to survey other methods that might be encountered.

During Class:

- In-Class Demonstration: Completion of an implicit 2-D finite element code
- Finite Volume Method on Regular Grids (cell-based)
- Mimetic Method for Diffusion on Irregular Grids
- *Production Essentials Lecture:* User Support, Bug Reporting, Issue Tracking

  Project (10%): Project status update

6.13 Week #13 Other Types of PDEs and Advanced Testing (week of 4/10)

This week, we transition away from diffusion-type partial differntial equations and focus on a different type of partial differential equations, one that is hyperbolic instead of elliptic, representing shock physics.

During Class:

- Rankine-Hugoniot Equations
- Shock Physics and Artificial Viscosity
- In-Class Demonstration: Development of a 1-D shock physics code using the finite volume method
- *Productions Essentials Lecture:* Advanced Testing: Software Test Harnesses, Methods of Unit Testing, Unit Test Platforms
- *Production Essentials Lecture:* Communication (Style Guides, Programmer's Manuals, Programmer Training)

6.14 Week #14 Advanced Testing and Communication (week of 4/17)

This week we focus on advanced testing in a production environment by demonstrating the addition of a unit test system to a computational physics code and adding sophisticated, high-level convergence regression tests.

During Class:

- In-Class Demonstration: Adding a unit test to the regression test suite
- Numerics-based Verification: Stabiity, spatial and temporal convergence analysis and testing
- Numerics-based Verification: Matrix condition, performance-based testing
- In-Class Demonstration: Adding a convergence regression test

Assignment (10%): HW 6 Adding convergence testing to a regression suite

6.15 Week #15 Parallel Computing (week of 4/24)

Production, large-scale computational physics often involves parallel, distributed memory computing. This week focuses on a cursory view of parallel computing with a demonstration of one way to parallelize a diffusion solver.

During Class:

- Essentials of parallel computing
- In-Class Demonstration: Parallel diffusion solver
- Measuring performance
- Parallel file I/O

6.16 Week #16 Presentations and Course Recapitulation (week of 5/1)

Students will give presentations on their projects and a course recapitulation will be provided. The recapitulation will cover every element that will be included on the Final Exam.

During Class:

- Course recapitulation and review for Final Exam
- Students present project results

Project (10%): Project presentations

6.17 Week #17 Placeholder (check Final Exam date) (week of 5/8)

This week might be reserved for the Final Exam. Check the schedule to be sure. The final exam might be on Saturday prior.

During Class:

- No lectures.

Exam (15%): Final Exam, in class, closed book

Project (5%): Project Final Report

7 Estimated Schedule at a Glance

Note that the schedule and content are subject to change, and in the table, below,

- "HW" = Homework assignment
- "MS" = Project milestone

Week	Nominal Topic	Assignments and Project Elements Mentioned
1/16	Foundations (Part 1)	HW: HW 1 Programming basics
1/23	Foundations (Part 2)	HW: HW 2 Math primer and governing equations
1/30	Foundations (Part 3)	
2/6	Regression Testing of a Single Physics Code	MS: Written project pre-proposal
2/13	Self-Documenting Regression Testing	MS: Revised project proposal
2/20	Preparing for Multi-Physics Code Development	MS: Project plan presentations and peer review.
2/27	Multi-Physics and Introduction to MMS	HW: HW 3 Nonlinear solver for multiphysics code
3/6	Code Verification	HW: HW 4 Adding MMS to a multi-physics code
3/13	Other Linear Solvers	
3/20	Other Discretization Methods: Finite Elements	HW: HW 5 Finite elements theory
3/27	Spring Break	
4/3	Other Discretization Methods: Mimetic and Finite Volume	MS: Project status update
4/10	Other Types of PDEs and Advanced Testing	
4/17	Advanced Testing and Communication	HW: HW 6 Adding convergence testing to a regression suite
4/24	Parallel Computing	
5/1	Presentations and Course Recapitulation	MS: Project presentations
5/8	Placeholder (check Final Exam date)	Exam: Final Exam, in class, closed book MS: Project Final Report

8 Policies

Classroom Behavior

Both students and faculty are responsible for maintaining an appropriate learning environment in all instructional settings, whether in person, remote or online. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. For more information, see the classroom behavior policy, the Student Code of Conduct, and the Office of Institutional Equity and Compliance.

Requirements for COVID-19

As a matter of public health and safety, all members of the CU Boulder community and all visitors to campus must follow university, department and building requirements and all public health orders in place to reduce the risk of spreading infectious disease. CU Boulder currently requires COVID-19 vaccination and boosters for all faculty, staff and students. Students, faculty and staff must upload proof of vaccination and boosters or file for an exemption based on medical, ethical or moral grounds through the MyCUHealth portal.

The CU Boulder campus is currently mask-optional. However, if public health conditions change and masks are again required in classrooms, students who fail to adhere to masking requirements will be asked to leave class, and students who do not leave class when asked or who refuse to comply with these requirements will be referred to Student Conduct and Conflict Resolution. For more information, see the policy on classroom behavior and the Student Code of Conduct. If you require accommodation because a disability prevents you from fulfilling these safety measures, please follow the steps in the “Accommodation for Disabilities” statement on this syllabus.

If you feel ill and think you might have COVID-19, if you have tested positive for COVID-19, or if you are unvaccinated or partially vaccinated and have been in close contact with someone who has COVID-19, you should stay home and follow the further guidance of the Public Health Office (contacttracing@colorado.edu). If you are fully vaccinated and have been in close contact with someone who has COVID-19, you do not need to stay home; rather, you should self-monitor for symptoms and follow the further guidance of the Public Health Office (contacttracing@colorado.edu). There is no requirement to notify the instructor that you will be missing lecture or have missed lecture due to COVID-19, any other illness, or any other reason.

Accommodation for Disabilities

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your faculty member in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities in the academic environment. Information on requesting accommodations is located on the Disability Services website. Contact Disability Services at 303-492-8671 or dsinfo@colorado.edu for further assistance. If you have a temporary medical condition, see Temporary Medical Conditions on the Disability Services website.

Preferred Student Names and Pronouns

CU Boulder recognizes that students’ legal information doesn’t always align with how they identify. Students may update their preferred names and pronouns via the student portal; those preferred names and pronouns are listed on instructors’ class rosters. In the absence of such updates, the name that appears on the class roster is the student’s legal name.

Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the Honor Code. Violations of the Honor Code may include, but are not limited to: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access to academic materials, clicker fraud, submitting the same or similar work in more than one course without permission from all course instructors involved, and aiding academic dishonesty. All incidents of academic misconduct will be reported to Student Conduct & Conflict Resolution (honor@colorado.edu); 303-492-5550). Students found responsible for violating the Honor Code will be assigned resolution outcomes from the Student Conduct & Conflict Resolution as well as be subject to academic sanctions from the faculty member. Additional information regarding the Honor Code academic integrity policy can be found on the Honor Code website.

Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

CU Boulder is committed to fostering an inclusive and welcoming learning, working, and living environment. University policy prohibits sexual misconduct (harassment, exploitation, and assault), intimate partner violence (dating or domestic violence), stalking, protected-class discrimination and harassment, and related retaliation by or against members of our community on- and off-campus. These behaviors harm individuals and our community. The Office of Institutional Equity and Compliance (OIEC) addresses these concerns, and individuals who believe they have been subjected to misconduct can contact OIEC at 303-492-2127 or email cureport@colorado.edu. Information about university policies, reporting options, and support resources can be found on the OIEC website.

Please know that faculty and graduate instructors have a responsibility to inform OIEC when they are made aware of any issues related to these policies regardless of when or where they occurred to ensure that individuals impacted receive information about their rights, support resources, and resolution options. To learn more about reporting and support options for a variety of concerns, visit Don't Ignore It.

Religious Holidays

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance.

See the campus policy regarding religious observances for full details.

Compiled Thursday 4th May, 2023 at 17:21